PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 13/00, 13/24, 15/16 — A1

(11) International Publication Number: WO 96/02036

(43) International Publication Date: 25 January 1996 (25.01.96)

(21) International Application Number: PCT/US95/08603

(22) International Filing Date: 7 July 1995 (07.07.95)

(30) Priority Data:
08/273,001 7 July 1994 (07.07.94) US

(71) Applicant: ELONEX TECHNOLOGIES, INC. [US/US]; Suite 110, 430 N. Mary Avenue, Sunnyvale, CA 94086 (US).

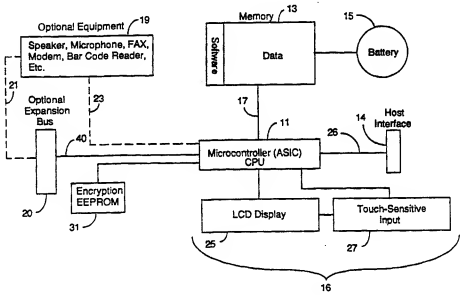
(72) Inventors: KIKINIS, Dan; 20264 Ljepava Drive, Saratoga, CA 95070 (US). DORNIER, Pascal; 374 North Murphy Avenue, Sunnyvale, CA 94086 (US). SEILER, William, J.; 2420 Glen Canyon Road, Scotts Valley, CA 95066 (US).

(74) Agent: BOYS, Donald, R.; P.O. Box 187, Aromas, CA 95004 (US).

(81) Designated States: CN, JP, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: MICRO PERSONAL DIGITAL ASSISTANT



(57) Abstract

A personal digital assistant module (10) with a local CPU (11), memory (13), and I/O interface (16) has a host interface (14) comprising a bus (26) connected to the local CPU (11) and a connector at a surface of the personal digital assistant (10) for interfacing to a bus connector of a host general-purpose computer, providing direct bus communication between the personal digital assistant (10) and the host general-purpose computer. In an embodiment, the personal digital assistant (10) also has a means for storing security code. The personal digital assistant (10) according to the invention forms a host/satellite combination with a host computer having a docking bay (63), wherein upon docking a docking protocol controls access by the host to memory (13) of the personal digital assistant (10) based on one or more passwords provided by a user to the host. In another embodiment the personal digital assistant (10) also has an expansion port (20) connected to the local CPU (11), and expansion peripheral devices may be connected and operated through the expansion port (20).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	**GB**	United Kingdom	**MR**	Mauritania
AU	Australia	**GE**	Georgia	**MW**	Malawi
BB	Barbados	**GN**	Guinea	**NE**	Niger
BE	Belgium	**GR**	Greece	**NL**	Netherlands
BF	Burkina Faso	**HU**	Hungary	**NO**	Norway
BG	Bulgaria	**IE**	Ireland	**NZ**	New Zealand
BJ	Benin	**IT**	Italy	**PL**	Poland
BR	Brazil	**JP**	Japan	**PT**	Portugal
BY	Belarus	**KE**	Kenya	**RO**	Romania
CA	Canada	**KG**	Kyrgyzstan	**RU**	Russian Federation
CF	Central African Republic	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CG	Congo			**SE**	Sweden
CH	Switzerland	**KR**	Republic of Korea	**SI**	Slovenia
CI	Côte d'Ivoire	**KZ**	Kazakhstan	**SK**	Slovakia
CM	Cameroon	**LI**	Liechtenstein	**SN**	Senegal
CN	China	**LK**	Sri Lanka	**TD**	Chad
CS	Czechoslovakia	**LU**	Luxembourg	**TG**	Togo
CZ	Czech Republic	**LV**	Latvia	**TJ**	Tajikistan
DE	Germany	**MC**	Monaco	**TT**	Trinidad and Tobago
DK	Denmark	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**US**	United States of America
FI	Finland	**ML**	Mali	**UZ**	Uzbekistan
FR	France	**MN**	Mongolia	**VN**	Viet Nam
GA	Gabon				

Micro Personal Digital Assistant

Field of the Invention

This invention is in the area of portable computers and pertains more specifically to small portable computing devices known in the art as personal digital assistants.

Cross Reference to Related Documents

This patent application is a continuation-in-part of copending patent application S/N 08/144,231, and of copending patent application 08/086,719.

Background of the Invention

Personal Digital Assistant (PDA) units, as of the date of this disclosure, enjoy a position of hope in the computer marketplace. Some believe this approach, a small, relatively inexpensive, and eminently portable computer unit, having software specifically written for tasks a user might expect to perform while travelling, will provide eminently useful and therefore salable computer products. Apple Computer, Hewlett Packard, and several other well-known computer manufacturers have made a considerable investment at no small risk in such systems.

Given the new systems now introduced, and those coming, for what is now known about them, there are still a number of drawbacks and problems. For example:

1. The PDA systems introduced are relatively costly, with starting prices ranging from several hundred dollars to two thousand

dollars and more. At such prices, rivalling current pricing for desktop systems, the buying public may react negatively. It is true that prices will fall with increased manufacturing volume and competition, but the high end start may well be rejected by potential users.

2. The systems being offered are still relatively bulky, considering the limited range of tasks that may be accomplished. Most are certainly too big to be conveniently carried in a breast pocket. The Newton, manufactured by Apple Corporation, weighs about a pound and is approximately the size of a VHS video cassette.

3. A big drawback of the PDA systems being offered is the way they transfer data between a user's desktop unit, or other host, and the PDA. Known communication is by modem, by infrared communication, and by serial connection. These all require manipulation by a user, modulation on one or both ends of the communication path, and the like, which can be time-consuming, error-prone, and hardware extensive (expensive). Presently the Newton offers a modem and/or LED communication as an option, adding to the overall cost.

4. In known PDAs, software is typically recorded in ROM, so updating applications can be difficult, and sometimes impossible. This will be a problem because PDA users will not want the PDA to have the same capabilities at all times. Typical users will be people who travel and work while they travel. These users require different functions for a trip to Taiwan than for a trip to France, for example. What is needed is a quick and convenient means to update and substitute software.

5. Another difficulty is in the fact that the data files a user manipulates while travelling are typically data files also resident in a home unit, herein called a host unit, such as the user's office desktop machine or notebook or other portable computer. It is very

troublesome to have two or more sets of critical data, with differences that one must remember to correct at an appropriate time. This can cause unending grief if files are not correctly updated. At best, current PDAs must use a relatively slow compressed bus to download and upgrade files. Typically this is done through a serial port, using a linking application like Laplink™.

What is needed is a small and inexpensive PDA that has a range of features that eliminate the above-described risks and problems. This new unit needs to be smaller than those presently being introduced, such as about credit-card size, or perhaps modeled on the PCMCIA type II or type III standard form factors. It should be inexpensive enough to produce that at least a minimum version could be sold in the roughly $100-$200 range, so it will be a unit seen to be a relatively inexpensive necessity. A PDA unit of this sort is the subject of the present invention, and is termed by the inventors a micro-PDA, or μPDA.

A very important feature of the μPDA in an aspect of the present invention is a direct parallel bus interface with a connector allowing the unit to be docked by plugging it into a docking bay in a host unit. Moreover, when the μPDA is docked in the host, there needs to be a means to effectively disable the CPU in the μPDA and to provide direct access to both the μPDA software and data storage by the host CPU. This direct access would provide immediate ability to communicate in the fastest available fashion between the μPDA and the host, and would also facilitate additional important features to be described below.

The μPDA also needs to have an optional compressed bus interface, including a connector separate from the host interface, so add-on devices may be utilized, such as a FAX modem, cellular communication, printer, and so on.

An additional feature that could be optionally provided in

another aspect of the invention is an interface at the host to allow a user to select pre-arranged software mixes for loading to the μPDA. This feature comprises a set of control routines operating in conjunction with the host's display and input means, to allow the user to quickly select applications and perhaps data as well to be loaded to the μPDA satellite, to configure the smaller, more portable unit for specific itineraries and purposes.

Another desirable feature is an ability to automatically update data files. In this aspect of the invention, with the μPDA docked, data on the host, if carrying a later date and/or time stamp than the data on the μPDA, would be automatically updated on the μPDA and vice-versa. When one returns from an excursion using the μPDA and docks the satellite at the host, the host gains access. determines the location of the latest files, and accomplishes the update. This feature needs to have some built-in user prompting to be most effective. It makes the μPDA a true satellite system.

Summary of the Invention

In a preferred embodiment of the invention a personal digital assistant module is provided comprising an enclosure for enclosing and supporting internal elements, a microcontroller within the enclosure for performing digital operations to manage functions of the personal digital assistant module, and a memory means connected to the microcontroller by a memory bus structure for storing data and executable routines. There is a power supply means within the enclosure for supplying power to functional elements of the personal digital assistant module, a display means operable by the microcontroller and implemented on a surface of the enclosure, and input means connected to the microcontroller for providing commands and data to the personal digital assistant module. A host interface

means comprising a host interface bus structure, which may be configured as a PCMCIA bus interface, is connected to the microcontroller and to a first portion of a host interface connector at a surface of the enclosure, and the host interface means is configured to directly connect the microcontroller to a compatible bus structure of a host computer.

In one embodiment the personal digital assistant module has an expansion bus interface comprising an expansion bus structure connected to the microcontroller and to a first portion of an expansion bus connector for connecting the microcontroller to a peripheral device. A wide variety of peripheral devices are provided for use with the personal digital assistant of the invention.

In another aspect, the personal digital assistant module also has a nonvolatile storage device, such as an EEPROM connected to the microcontroller and containing one or more codes unique to the personal digital assistant, for uniquely identifying the personal digital assistant to digital devices connected on the host interface.

In a preferred embodiment, the display and input means for the personal digital assistant are configured as an overlaid touch screen and LCD display on a surface of the outer case of the personal digital assistant. A pointer device implemented as a thumbwheel in one embodiment and as a pressure sensitive pad in another is provided as part of the input capability.

The personal digital assistant module forms a unique combination with a general-purpose computer host having the personal digital assistant as a satellite unit. The host in this instance has a docking bay especially configured to dock the personal digital assistant, making a direct bus connection between the local CPU of the personal digital assistant and the CPU of the host. The host may be a desktop unit, a notebook computer, or a smaller portable like a

palmtop computer. This combination provides power and convenience not before available.

Many other digital devices are also provided according to various aspects of the invention, such as modems, scanners, data acquisition peripherals, cellular phones, and a software vending machine, and all of these devices may be appended to the personal digital assistant by the expansion bus interface or, in many cases, by the host interface.

The personal digital assistant provided according to embodiments of the present invention is a unit more compact than conventional PDAs. It represents a new dimension in computer application and applicability, in a form promising to be eminently usable by and useful to almost everyone; and at a price easily affordable. It solves the communication problem intrinsic to personal digital assistants relative to larger and more powerful computers, with a unit that fits into a user's breast pocket, and at a very low price.

Brief Description of the Drawings

Fig. 1A is an isometric view of a μPDA according to an embodiment of the present invention.

Fig. 1B is a plan view of the μPDA of Fig. 1A.

Fig. 2 is a cross-sectional view of the μPDA of Figs. 1A and 1B.

Fig. 3 is a block diagram of the μPDA of Fig. 1A and some peripheral elements.

Fig. 4 is a more detailed plan view of the μPDA of Fig. 1A showing in particular an LCD display and touch screen user interface in an aspect of the present invention.

Fig. 5 is an isometric view of a μPDA and a host notebook computer in an aspect of the present invention, with the μPDA about

to be docked in a docking bay of the notebook computer.

Fig. 6 is a block diagram of a μPDA docked in a docking bay of a host computer according to an embodiment of the present invention.

Fig.7 is a logic flow diagram of the steps in docking a μPDA in a host computer according to an embodiment of the present invention.

Fig. 8 is an isometric illustration of a μPDA software vending machine in an aspect of the present invention.

Fig. 9 is a top plan view of a μPDA enhanced user interface according to an embodiment of the present invention.

Fig. 10 is a top plan view of a μPDA with a microphone in an embodiment of the present invention.

Fig. 11 is an isometric drawing of a μPDA docked in a dedicated cellular or cordless telephone according to an embodiment of the present invention.

Fig. 12 is a plan view of a μPDA with a speaker and pager interface according to an embodiment of the present invention.

Fig. 13 is a plan view of a μPDA with an infrared communication interface according to an embodiment of the present invention.

Fig. 14 is a plan view of a μPDA with a scanner attachment according to an embodiment of the present invention.

Fig. 15 is a plan view of a μPDA with a fax-modem attached according to an embodiment of the present invention.

Fig. 16 is a plan view of a μPDA with a printer adapter interface according to an embodiment of the present invention.

Fig. 17 is an isometric drawing of a μPDA docked in a barcode reader providing a data acquisition peripheral according to an embodiment of the present invention.

Fig. 18 is an isometric view of a μPDA with a solar charger according to an embodiment of the present invention.

Fig. 19 is a plan view of four μPDAs interfaced to a dedicated network console providing inter-PDA communication according to an embodiment of the present invention.

Fig. 20 is an isometric view of a μPDA according to the invention connected by the expansion port to a standard-sized keyboard.

Fig. 21 is a block diagram illustrating a computer architecture according to one embodiment of the invention.

Fig. 22 is a block diagram illustrating a computer architecture as shown in Fig. 21, comprising also an optional interface and cache system.

Fig. 23 is a pin-out listing of the optimized bus structure of the invention in a preferred embodiment.

Fig. 24A illustrates a memory mapping scheme for a general-purpose computer according to an embodiment of the present invention.

Fig. 24B illustrates a bus controller incorporating an IRQ controller and a shadow DMA controller according to an embodiment of the invention.

Fig. 25 is a timing diagram showing the states of the Clock, HCI Bus Address/Data, Address Strobe, Device Decode, Ready, and Read/Write lines during back-to-back read operations in an embodiment of the invention.

Fig. 26 is a timing diagram showing the bus states for a burst-mode read operation, also called a HCI master-mode read, in an embodiment of the present invention.

Fig. 27 is a timing diagram showing the HCI bus states for back-to-back write cycles in an embodiment of the invention.

Fig. 28 is a timing diagram showing the HCI bus states for a burst-mode write transfer in an embodiment of the present invention.

Description of the Preferred Embodiments

Fig. 1A is an isometric view of a μPDA 10 according to an embodiment of the present invention. In this embodiment the unit is modeled on the PCMCIA standard Type II form factor, having a height D1 of about 5mm. Body 12 is described in further detail below, and has a female portion 14 of a connector recessed at one end for engaging a mating male portion of the connector in a host computer, connecting the μPDA internal circuitry directly with a host internal bus. The host unit may be a notebook computer having a docking bay for the μPDA. Docking bays may be provided in desktop and other types of computers, and even in other kinds of digital equipment, several examples of which are described below.

Still referring to Fig. 1A, in this embodiment there is a combination I/O interface 16 implemented on one side of the μPDA, comprising a display overlaid with a touch-sensitive planar structure providing softkey operation in conjunction with interactive control routines operable on the μPDA in a stand-alone mode.

Although not shown in Fig. 1A, there may also be guides implemented along the sides of the case of the device for guiding the module in and out of a docking bay in a host computer unit. There may also be one or more mechanical features facilitating engagement and disengagement of the module in a docking bay.

Fig. 1B is a top plan view of the μPDA of Fig. 1A, showing a thumbwheel 18 implemented in one corner of the μPDA. The thumbwheel in this embodiment is an input device capable of providing input with both amplitude and directional characteristics, and in some cases rate characteristics as well. The thumbwheel has many uses in combination with the μPDA and I/O interface 16. One such use is controlled scrolling of icons, characters, menus, and the like on the display of the device. The thumbwheel provides many of

the functions of a pointer device.

In this embodiment of the μPDA a second external connector portion 20 is provided. This connector portion is for engaging peripheral devices as part of an expansion bus interface.

Fig. 2 is a simplified cross-sectional view of a means for constructing a μPDA according to the present invention in a Type II PCMCIA, or other relatively small package. ICs 34 are encapsulated in a conformal material 36, and interconnection is accomplished by traces on a flexible polymer film 32 shown as overlaying the encapsulated structure. In this structure the ICs are not packaged in the conventional manner having solder leads for assembly to a printed circuit board. Rather, connections are made directly between the solder pads on the chip and the traces on the Kapton film. Also there is no intention to relate ICs indicated by element No. 34 with specific functional ICs in a μPDA. This cross-section is illustrative of a method of construction only.

In this compact construction there may also be traces on the side of film 32 away from the interconnections for the CPU and memory for connection to other elements, such as display 25 and touch-sensitive screen 27.

LCD display 25 is implemented on one side of the μPDA, and touch-sensitive interface 27 is provided overlaying at least a portion of the LCD display. A metal casing 38, or other suitable material or combinations of material, surrounds the internal components and conforms to Type II PCMCIA form factors. This simplified cross-section illustrates some of the principles of construction that can allow the needed components to be inexpensively fitted into the small form factor needed. In another embodiment the μPDA is implemented in the form factor of a type III (10 mm thick) PCMCIA unit, using relatively conventional technology, such as PCB technology, rather than the encapsulated construction described

immediately above. Various other constructions, form factors, and combinations are possible, as well.

Fig. 3 is a simplified electrical block diagram of the μPDA of Figs. 1A, 1B and 2. A unique microcontroller 11 acts as the CPU of the μPDA in the stand-alone mode, that is, when the μPDA is not docked in a host unit. When the μPDA is docked in a host computer, microcontroller 11 acts as a slave unit, granting bus control to the CPU of the host. In docked mode, the CPU of the host thus gains control of the memory contents of the μPDA, subject in most cases to security procedures which are described below. Thus the host computer can transfer data and software into and out of a docked μPDA memory. In other embodiments many other cooperative operating modes may be accomplished between the two CPUs and accessible memory devices.

Memory 13 is preferably a nonvolatile device from 1 to 2 megabytes in this embodiment, and both control routines for applications and data files are stored in this memory. Memory 13 may be flash memory, CMOS ROM, CMOS RAM with battery, or a combination, with the software stored in ROM and the data in the flash memory. The memory device is interfaced to microcontroller 11 via a dedicated bus structure 17, and microprocessor 11 is configured to drive memory bus 17.

A battery 15 is the power source in the stand-alone mode, and may be recharged in one or more of several ways. The power traces are not shown in Fig. 3, but extend to all of the powered devices in the μPDA module. When the unit is docked in the host, the host power source may be connected to pins through the host interface to recharge the battery. Alternatively, an attached means such as a solar panel may be configured to charge the battery and/or provide power to the μPDA. A solar panel for power is described elsewhere in this disclosure. Also the battery may be easily removed for periodic

replacement.

Host bus connector 14 is a part of a host interface which comprises a bus structure 26 for providing connection to the host in docked mode, as described above. In a preferred embodiment, the host interface is according to PCMCIA Type II, Rev. 3 standard, which is capable of communication either in PCMCIA mode or in a mode similar to PCI mode. PCI mode refers to a high-speed intermediate bus protocol being developed by Intel corporation, expected to become a standard bus architecture and protocol in the industry. The physical interface at the host in this embodiment is a slot-like docking bay, as is typical of know docking bays for PCMCIA devices. This docking bay may be implemented as a docking box, a built-in unit like a floppy-drive unit, or it may take some other form.

Connector portion 20 is a part of the expansion bus interface described above, comprising a dedicated bus structure 40 connected to microcontroller 11. This interface can be implemented in a number of different ways. The purpose of the optional expansion bus interface is to connect to optional peripheral devices, such as a printer, a FAX modem, a host cellular phone, and others. The expansion bus interface is not an essential feature in a minimum embodiment of the present invention, but provides vastly enhanced functionality in many embodiments.

The expansion interface can take any one of several forms. A preferred form is an extended enhanced parallel port and protocol based on an invention by the present inventors disclosed in a copending patent application. Another form is an indexed I/O port having 8-bit address and 8-bit data capability. The requirement of the expansion port is that the connection and communication protocol be compatible with expansion devices, such as telephone modems, fax modems, scanners, and the like. Many other configurations are

possible.

Optional equipment such as devices listed in box 19 may be connected for use with the μPDA through the expansion bus. Selected ones of such devices may also be built in to the μPDA in various embodiments, providing variations of applicability. In the former case, connection is through path 21 and the expansion bus interface via connector portion 20. In the built-in case, connection is in the interconnection traces of the μPDA as indicated by path 23.

I/O interface 16 (also Fig. 1B) is for viewing μPDA application-related data and for touch-sensitive input via softkeys. By softkeys is meant assignment by software of various functions to specific touch sensitive screen areas, which act as input keys. Labels in I/O interface 16 identify functionality of the touch-sensitive areas in various operating modes according to installed machine control routines. LCD display 25 and the touch-sensitive area 27 together form the combination I/O interface 16 described also above.

In some embodiments of the present invention, data and program security is provided comprising an Electrically Erasable Programmable Read Only Memory (EEPROM) 31, which is connected by dedicated communication lines to microcontroller 11. EEPROM 31 holds one or more codes installed at the point of manufacturing to provide security for information transfer between a host and a μPDA. The purpose is to control access by a host to the memory contents of a μPDA, so each μPDA may be configured to an individual. To accomplish this, docking and bus mastering machine control routines are initiated at the point of docking, and this security process is described in more detail below. In other embodiments, security codes may be provided by a Read Only Memory (ROM) chip or other permanent or semi-permanent memory source.

Fig. 4 is a plan view similar to Fig. 1B, of a μPDA, showing in particular I/O interface 16. The size and location of I/O interface

16 may vary, but in general occupies a major portion of one of the sides of the module. In one embodiment I/O interface 16 comprises an LCD display with a resolution of 256 by 144 pixels in a screen size that displays 32 by 12 characters. Each character in this embodiment is displayed in an area eight pixels wide and twelve pixels high. In another embodiment, the pixel resolution is 320 by 200, which corresponds to 40 by 16 characters.

The touch-sensitive areas of the touch-sensitive screen correspond to the character areas of the display. By touching an area with a finger or stylus, data can be entered quite quickly and with minimal CPU demand.

At one corner, thumbwheel 18 provides a two-directional means of controlling the configuration of the display according to installed control routines. A menu 70 is configured at one side to represent the current status of any application in progress and to provide appropriate user menu selections. In a preferred embodiment input from thumbwheel 18 is used for scrolling through menu 70, and active areas may be indicated by a cursor. A user makes a menu selection by pressing the appropriate touch-sensitive area. A specific input may be provided to cause the menu area to be displayed on either side of the display according to a user's preference.

Specific characters are displayed in this embodiment in a region 74, with each character area associated with a touch-sensitive input area. As region 70 dedicated to selectable characters is much too small to display all characters of a standard keyboard, input from thumbwheel 18 allows a user to pan region 74 displaying an entire virtual standard keyboard. Movement of thumbwheel 18 in one direction pans the character region horizontally, and movement in the other direction pans the character region vertically. When an end is reached the window pans onto the virtual keyboard from the other

end. In this manner, a user may quickly pan the character window to display an entire standard keyboard, and make selections with a finger or a stylus. Of course, it is not required that a virtual keyboard be laid out for access in the format of a standard keyboard. Characters and punctuation, etc., could just as simply be displayed in a single strip along a region of the display, and scrolled by input from the thumbwheel or other pointer-type input device.

In this embodiment, to avoid delays caused by panning, if the thumbwheel is rotated quickly the character window jumps rather than scrolling to speed up the interface. In addition, menu 70 may optionally provide for a character display in different fonts and sizes, although a single font is preferred to minimize memory demand. It will be apparent to those with skill in the art that there are many alternatives for character selection and display, and many ways thumbwheel 18 may be configured to allow for scrolling and panning.

A document window 72 is provided in this embodiment at the top or bottom of I/O interface 16. A cursor locates the active position within the document for editing purposes. Menu 70 provides selection of available fonts, and input by thumbwheel 18 controls cursor movement over the document. As a document will in almost all cases be much larger than the display capability of region 72, it is necessary to pan the document window in essentially the same manner as the keyboard window is panned. For example, rotating thumbwheel 18 in one direction may display horizontal strips of a document, while rotating the thumbwheel in the opposite direction moves the window vertically strips of the same document.

A soft key or optional hard key may be configured to switch between the document and keyboard window, and the same or another key may be configured to switch between scrolling left or right, up or down, document or keyboard. A switch key may be used to change

the thumbwheel mode of operation. A switch key may also be used in combination with a floating pointer to select characters and menu items. In this embodiment, the user can keep his or her hands relatively stationary on just the thumbwheel and the switch key, making all possible selections. Use of a a switch key in combination with a floating pointer facilitates the use of small fonts. A switch key may also be incorporated as an additional hard key in a convenient location on the case 12.

It will be apparent to a person skilled in the art than there are numerous ways to combine menu selections, switching keys and I/O configurations to provide a user-friendly user interface. A further embodiment of the present invention provides an I/O set-up application wherein a user may completely customize features of I/O area displays.

There are other sorts of mechanical interfaces which may be used to provide pointer-style input in different embodiments of the invention as alternatives to the thumbwheel disclosed. One is a four-way force-sensitive mouse button and a selector button, which may be located at opposite ends of case 12 below I/O interface 16. Each button is designed to be operated by one finger. The four-way force-sensitive mouse button can provide menu scrolling of a cursor and panning and/or indexing of keyboard and document windows, while the selector button is used to select and edit according to position of a cursor. This configuration minimizes hand movement and keeps the I/O area clear for viewing.

Implementation of thumbwheels, pressure-sensitive switches and buttons, and the like, are known in the art, including the translation of mechanical motion and pressure to electrical signals and provision of such signals to a microcontroller. For this reason, details of such interfaces are not provided in this disclosure. Combinations of

such inputs with displays and input areas may, however, be considered as inventive.

Fig. 5 is an isometric drawing of a μPDA 10 in position to be docked in a notebook computer 172 via a Type II PCMCIA docking port 105 according to an embodiment of the present invention. As further described below, once the μPDA is docked, it is activated and a procedure is initiated with the host computer to manage communication and verify memory access rights (security).

Access rights are considered important by the inventors for a number of reasons. Firstly, through the expedient of one or more specific codes, unique to each μPDA, a user may protect files stored in his module from access by unauthorized persons. The code can be used both to control access to data and files via I/O interface 16, and also through the host bus interface, so data and files may be secure from access by an unauthorized host system.

In the former case, when a μPDA is powered up, an application routine can query the user for an access code to be entered at I/O interface 16 Fig. 4). If the code is not entered properly, access is denied, and power goes off. Codes for the purpose are stored in EEPROM 31 (Fig. 3), or in whatever ROM device may be devoted to the purpose. In some embodiments, the code may by mask-programmed at manufacture, so it is not alterable. In others, the code may be accessible and changeable by special procedures in the field.

In the case of host communication, it is possible that a portable or desktop computer, or some other device, may have a docking port physically configured to receive a μPDA, yet not be configured to communicate with the μPDA. This certainly might be the case where the μPDA is in the PCMCIA form. For purposes of disclosure and description, this specification terms such a unit a generic host. If the unit is configured to communicate with a μPDA it is an enabled host. If a host is configured for full access to a particular μPDA, it is a dedicated host.

If a docking unit is a generic host, there will be no communication unless the person presenting the μPDA provides the control routines to the host. This may be done for a generic host such as by transfer from a floppy disk, from a separate memory card through the docking port, or, in some embodiments, the communication software may be resident in memory 13 (Fig. 3) of a docked μPDA, transferrable to the host to facilitate further communication.

If the docking unit is in fact an enabled host, or is configured after docking to be an enabled host, the stored code or codes in EEPROM 31 (or other storage unit) may be used to verify authorization for data and program transfer between the host and a μPDA. In one embodiment this procedure is in the following order: First, when one docks a μPDA in a compatible docking port, certain pin connections convey to both the μPDA microcontroller and to the host CPU that the module is docked. Assuming an enabled host, the fact of docking commences an initialization protocol on both systems.

In most embodiments, if the docking unit is a non-host, that is, it is not capable of communication with the docked module, nothing happens, and the user may simply eject the docked module. If the computer is an enabled host, an application is started to configure host access to the μPDA's data files through the μPDA microcontroller. A user interface, described more fully below for a particular embodiment, is displayed on the host monitor 104 (Fig. 5). The host interface menu, as well as other application menus, may be formatted in part as a display of the μPDA I/O interface 16 as seen in Fig. 4 and described in accompanying text. In some embodiments, the docked μPDA can be operated in situ by manipulating the input areas of the μPDA displayed on the host's screen.

If the host is not a home unit for the docked module, that is, the host does not have matching embedded ID codes to those stored in the docked module, a visitor protocol is initiated. In this event, a

visitor menu is displayed on host display 104 for further input, such as password queries for selections of limited data access areas in the docked module. In this case, too, a user may gain full access to the docked module's memory registers by entering the proper password(s).

If the host is a fully compatible host home unit, full access may be immediately granted to the host to access memory contents of the docked module, including program areas; and both data and programs may be exchanged.

In any case, when the μPDA is ejected or otherwise removed from the docking port, the on-board module microcontroller again gains full control of the internal μPDA bus structures.

Fig. 6 is a simplified block diagram of a μPDA docked in a host computer, and Fig. 7 is a basic logic flow diagram of the steps involved in docking a μPDA in a host computer 66 according to an embodiment of the present invention. Host computer 66 is represented in a mostly generic form, having a host CPU 24, and input device 60, such as a keyboard, a mass storage device 28, such as a hard disk drive, and system RAM 62. It will be apparent to those with skill in the art that many hosts may have a much more sophisticated architecture, and the architecture shown is meant to be illustrative.

When a μPDA unit is docked, connector 14' in Fig. 6 comprises portion 14 shown in Figs. 1B and 3 and a mating connector portion for engaging portion 14 in port 105 (Fig. 5). The engagement of the separate portions of the connector cause bus 26 in the μPDA and bus 26' in the host to become directly connected. There is then a direct bus path between microcontroller 11 and host CPU 24 (Fig. 6).

As previously described there is a pin configuration (not shown) in connector 14 dedicated to signalling that a module is docked. In Fig. 7, step 42 represents insertion of a μPDA module into the docking port. At step 44 the signalling pin configuration

signifies physical docking is accomplished. At step 46 host interface bus 26 is activated, including the mated host bus 26' in the host.

At step 48 (Fig. 7) microcontroller 11 in the μPDA starts a preprogrammed POST procedure. Microcontroller 11 in this embodiment has a page of RAM 68 implemented on the microcontroller chip. In other embodiments RAM may be used at other locations. At step 50, the POST routine loads a bootstrap program to RAM 68, which includes a code or codes for security matching. This code or codes comprise, for example, a serial number.

At step 54 the bootstrap program begins to execute in microcontroller 11, and at step 56 the microcontroller looks for a password from the host on host interface bus 26 (Fig. 6).

The fact of docking, assuming an enabled or dedicated host, also causes a communication routine, which may be accessed from, for example, mass storage device 28 at the host, to display a user interface on monitor screen 104 of the host unit, as partly described above. It is this communication program that makes a generic host an enabled host.

Assuming an enabled, but not dedicated, host, the user interface will query a user for input of one or more passwords, after successful entry of which the host will pass the input to microcontroller 11 for comparison with the serial number and perhaps other codes accessed from EEPROM 31 in the bootstrap of the μPDA.

According to the codes passed from the host to the docked module, microcontroller 11 will allow full access to memory 31 at function 52, Fig. 7, for the host CPU, or limited access at some level at function 58, defined by received codes (or no matching code at all).

The access protocols and procedures allowing partial or direct access to μPDA memory 13 are relatively well known procedures in the art, such as bus mastering techniques, and need not be reproduced in detail here. In addition to simple comparison of codes, there are

other techniques that may be incorporated to improve the integrity of security in the communication between a μPDA and a host. For example, within the limitation of storage capacity of the EEPROM or other nonvolatile source, executable code might also be uploaded to onboard RAM 68, or code keys to be used with executable code from other sources, or relatively simple maps re-allocating memory positions and the like, so each μPDA may be a truly unique device.

There are additional unique features provided in one aspect of the invention as part of the communication routines introduced above. One such feature is automatic updating and cross-referencing of existing files and new files in both computers, under control of the host system, with the host having direct bus access to all memory systems. Auto-updating has various options, such as auto-updating by clock signature only, flagging new files before transfer, and an editing means that allows the user to review both older and newer versions of files before discarding the older in favor of the newer. This automatic or semiautomatic updating of files between the satellite and the host addresses a long-standing problem. The updating routines may also incorporate a backup option to save older files.

Another useful feature in host/μPDA communication is a means for a user to select and compose a mix of executable program files for downloading to a μPDA, either replacing or supplementing those executable routines already resident. A user can have several different program lists for downloading as a batch, conveniently configuring the applicability of a μPDA among a wide variety of expected work environments.

Such applications as databases, spreadsheets, documents, travel files such as currency converters, faxing and other communications programs, time clocks, address and telephone records, and the like, may comprise customized lists of user-preferred applications.

In another embodiment, an undocked μPDA can transfer data

via the optional expansion bus 40 (Fig. 3) directly to a host. In the special case of a μPDA user without access to a PCMCIA interface on his host (notebook or desk-top) computer, he or she can connect to a host via an auxiliary port on the host, such as a serial port, via the expansion bus interface. In this case, the μPDA still requests password(s) from the host, and controls access to its on-board memory according to the password(s) received.

The optional expansion interface may also be used in some embodiments while a μPDA is mastered by a host, wherein the host may effectively send data through the bus structures of the μPDA.

Additional Aspects and Features

Software Vending Machine:

In a further aspect of the invention, a Software Vending Machine with a very large electronic storage capacity is provided, wherein a μPDA user may dock a module and purchase and download software routines compatible with the μPDA environment.

Fig. 8 is an isometric view of such a vending machine 61 having a docking bay 63 for a μPDA, a credit card slot 65, and a paper money slot 67. A display 69 provides a user interface for reviewing and purchasing software from the vending machine, along with selector buttons such as button 71 along the sides of the display. In an alternative embodiment the display may also have a touch screen, and may, in some embodiments, emulate the μPDA I/O area on a larger scale.

In operation, a user may, in this embodiment, review software for sale simply by docking his μPDA unit in the vending machine and selecting from a menu on display 69. The menu may allow the user to browse all available applications, or list new applications since entered dates. The user can select certain applications, try them out,

at least in simulation, and then select applications to purchase.

The vending machine, once all the requirements are met, such as proper identification and payment, copies the selected application(s) to the memory of the μPDA, or, alternatively, to a floppy disk provided by either the user or the vending machine. In this case there is also a floppy disk drive 73 in the vending machine and a port 75 for dispensing formatted floppies for a customer to use in the disk drive. This mode is useful for the instances where a user's μPDA is loaded beyond capacity to receive the desired software, or the user simply wishes to configure the software mix himself from his or her own host computer.

There may also be provided a backup option so a user may instruct the vending machine to read and copy all or a selection of his files to one or more floppy disks before installing new files or data.

As described above, each user's μPDA includes an EEPROM or other storage uniquely identifying the μPDA by a serial number or other code(s), so the vending machine may be configured in this embodiment to provide the software in one of several modes.

A user may buy for a very nominal price a demo copy of an application, which does not provide full capability of the application, but will give the user an opportunity to test and become familiar with an application before purchase. Also, the user may buy a version of the same application, configured to the ID key of the μPDA to which it is loaded, and operable only on that μPDA. In another embodiment, the software is transferable between a family of keyed μPDAs, or has the ability to "unlock" only a limited number of times. In these cases, the applications would be sold at a lesser price than an unlocked version. The unlocked version works on any μ-PDA and/or host/μPDA system. The higher price for the unlocked version compensates for the likelihood of unauthorized sharing of the vended applications.

The vending machine could also offer a keyed version,

customized to operate only on the μPDA docked in the software vending machine, or upon a family of μPDAs. This keyed version is possible because of the individual and unique nature of each μPDA, which has, at a minimum, a unique serial number, and may also have other security programming, as described above, which allows a vending machine to prepare and download a customized copy of an application that will operate only on the particular module for which it is purchased.

There are a number of different means by which unique correspondence might be accomplished, as will be apparent to those with skill in the art. A standard version stored in the memory facility of a vending machine might be recompiled, for example, on downloading, using a unique code from the docked or identified μPDA as a key in the compilation, so only the specific μPDA may run the program by using the same unique key to sequence the instructions while running. The key for scrambling or otherwise customizing an application might also comprise other codes and/or executable code sequences stored uniquely in a μPDA.

In yet another aspect related to the vending machine, there is a printer outlet 77 which prints a hardcopy manual for the user. It is, of course, not necessary that the software vended be specific to the M-PDA. Applications may also be vended for other kinds of machines, and transported in the memory of the μPDA, or by floppy disk, etc. In this embodiment a non-μPDA user can acquire a wide assortment of software.

The software vending machine may also serve as an optional informational display center in such locations as airports, train stations, convention centers, and hotels. Upon inserting a μPDA a user may interface directly and upload current information including, but not limited to, local, national, and world news; stock quotes and financial reports; weather; transportation schedules; road maps; language translators; currency exchange applications; E-mail and other

direct on-line services.

A customized vending machine could be tailored to business travelers and allow fast access to pertinent information, allowing the user to download files to send via E-mail. In another aspect of the invention, the vending machines are linked to each other allowing users to send messages to associates travelling through locations of associated vending machines. Such dedicated μPDA E-mail is immediately downloaded to a specific μPDA as it is docked. The sender may have the associate's μPDA unique encoded key as identification, or some other dedicated identifying means for E-mail.

In another embodiment, as each business associate arrives at an airport, he or she may prompt the custom vending machine in that location via an optional installed infrared interface (not shown) in their μPDA. The custom vending machine, also equipped for infrared communication, receives the signal and sends/or receives any messages that are waiting.

Enhanced Display:

Fig. 9 is a plan view of an enhanced I/O interface unit 79 according to an aspect of the present invention. Interface unit 79, with about a 5-inch diagonal measurement, comprises a combination LCD display at least partially overlaid by a touch-sensitive input screen, providing an I/O area 80 in much the same manner as in a μPDA. Four docking bays 81, 83, 85, and 87 are provided in the left and right edges of interface unit 79 in this embodiment, and are configured for PCMCIA type II modules. One of these bays may be used for docking a μPDA according to the present invention, and the other three to provide a larger CPU, additional memory, battery power, peripheral devices such as modems, and the like by docking functional PCMCIA modules.

Interface unit 79 is a framework for assembling a specialty computer through docking PCMCIA units, including a μPDA according to the present invention. In other embodiments where the μPDA assumes other form factors, the docking bays may be configured accordingly.

A docked μPDA in this embodiment is configured to produce its I/O display on I/O area 80. The thumbwheel on the M-PDA is accessible while docked and acts as described above in the stand-alone mode in this case. In another aspect, the enhanced display has a re-configured output that enables the user to manipulate the data from the touch-screen alone and/or additional hardware selector buttons and/or a standard keyboard attached to the enhanced display via a dedicated bus port, or even through the expansion port of a docked μPDA. In a further embodiment the enhanced display has a dedicated mouse port and/or a dedicated thumbwheel.

In yet another embodiment, interface unit 79 has an inexpensive, conventional, replaceable battery and/or a rechargeable battery. Also, in another aspect, interface unit 79 may dock two or more individual μPDAs and cross-reference data files between them according to control routines that can manipulate mutually unlocked files. Further still, interface unit 79 may be placed and structurally supported for easy viewing on a dedicated standard or smaller-sized keyboard, connecting to the keyboard as an input device. The keyboard would then automatically serve as the input device.

Interface unit 79 for a μPDA is small and compact enough to slip into a pocket book or briefcase, providing a very portable, yet very powerful, computer.

Microphone/Voicenotes:

Fig. 10 is a plan view of a μPDA 110 with an I/O interface

116, an expansion port 120, and a host interface connector 114. μPDA 110 has all the features previously described and additionally a microphone 88. In this embodiment, control routines in the μPDA use a linear pedictive coding (LPC) approach to convert analog input from the microphone to a digital voice recording. This approach uses a minimum of memory, but still is capable of reproducing audio input like the human voice within recognizable limits.

In an alternative embodiment, for better quality voice recording, a two-step integrator may be used in order to separate the analog signal and synthesize a closer digital representation.

With a μPDA so configured, a user's voice notes can be recorded and later uploaded to a host for processing. In future embodiments the digital signals may be converted to text or sent as voicemail on a network. In yet another embodiment, the microphone is integrated with a speaker for editing purposes.

Cellular Telephone Interface:

Fig. 11 is an isometric view of a μPDA 10 docked in a dedicated cellular telephone 45 according to an embodiment of the present invention. Telephone 45 has a docking port 49 for a μPDA according to the invention. In this embodiment, port 49 is on one side of telephone 45, and there is a window 51 to provide access to I/O interface 16 of the μPDA after it is docked. With the μPDA docked, all of the software and memory of the μPDA is available to the telephone and a user may operate the phone by I/O interface 16.

In this aspect of the invention, unique control routines and display configurations are provided to enhance use of the cellular phone. For example, all of the user's collection of phone numbers, associated credit card numbers, access codes, etc. are readily available and may be quickly and conveniently accessed and used. In one

aspect, a simple input displays alphabet letters to select, and once a letter is selected, a partial list of parties that might be called is displayed. One may scroll through the list by touch input or by use of the thumbwheel of the μPDA and select a highlighted entry. It is not required that the telephone numbers be displayed.

Once a party to be called is selected, the μPDA dials the call, including necessary credit card information stored in the memory of the μPDA for this purpose.

In a further embodiment, the calls are timed and time-stamped and a comprehensive log, with areas for notes during and after, is recorded.

In another embodiment, conversations are digitally recorded and filed for processing later. A future embodiment may include a voice compression program at a host or within cellular phone 45. Compressed voice files, such as, for example, messages to be distributed in a voicemail system, may be downloaded into the μPDA or carried in a larger memory format inside the cellular telephone. The μPDA can then send the files via a host or dedicated modem attached at connector portion 20 to the optional expansion bus 40 (Fig. 6).

The cellular telephone may, in this particular embodiment, have a bus port for digital transmission. In this case, the compression algorithm along with voice system control routines are also established at the receiving end of the transmission to uncompress the signal and distribute individual messages.

In a further embodiment, voice messages may be sent in a wireless format from the cellular telephone in uncompressed digital synthesized form, distributing them automatically to dedicated receiving hosts, or semi-automatically by manually prompting individual voicemail systems before each individual message. In a further aspect of wireless transmission, a microphone/voicenote μPDA

as in Fig. 10 may send previously stored voicenotes after docking in a cellular telephone interface.

In Europe and Asia a phone system is in use known as CT2, operating on a digital standard and comprising local substations where a party with a compatible cellular phone may access the station simply by being within the active area of the substation. In one aspect of the present invention, a CT2 telephone is provided with a docking bay for a μPDA, and configured to work with the μPDA. In yet another aspect of the invention, in the CT2 telephone system, and applicable to other digital telephone systems, a compression utility as disclosed above is provided to digitally compress messages before transmission on the CT2 telephone system.

It is roughly estimated that a dedicated compression algorithm may compress ten minutes of voice messages into one minute using the existing CT2 technology. This would save on telephone use charges significantly. In this aspect, there needs be a compatible decompression facility at the receiving station, preferably incorporated into a standard μPDA voicemail system for CT2 or other digital transmissions.

In a further embodiment, control routines are provided to enable the microphone/voicenote μPDA as illustrated in Fig. 10 to carry digital voicenotes, either compressed or uncompressed. When docked in a CT2-compatible μPDA cellular telephone, the μPDA in this embodiment can transmit the digital voicenotes in compressed form.

Speaker/Pager:

Fig. 12 is a plan view of a μPDA 210 with a microphone/speaker area 90 and a pager interface 92 according to an embodiment of the present invention. This μPDA has the ability to

act as a standard pager, picking up pager signals with installed pager interface 92 and alerting a user through microphone/speaker 90. Once the signals are received, μPDA 210 can be docked in a compatible cellular telephone as illustrated in Fig. 11 and the μPDA will automatically dial the caller's telephone number. All other aspects are as described in the docked mode in the cellular telephone.

In another embodiment, the speaker/pager μPDA can be prompted to generate DTMF tones. The DTMF tones are generated from a caller's telephone number.

The speaker/pager μPDA can store pager requests in its onboard memory. It can also display all pager requests including time and date stamps, identification of the caller, if known, and other related information, on I/O interface 216. In this particular embodiment, a user can receive a page, respond immediately in digital voicenotes on the μPDA via speaker/microphone 90, and then send the response from a dedicated μPDA-compatible cellular telephone or conventional telephone.

Wireless Infrared Interface:

Fig. 13 is a plan view of a μPDA 310 with an IR interface 94 according to an embodiment of the present invention. In this embodiment the μPDA may communicate with an array of conventional appliances in the home or office for providing remote control. Unique signals for the appliances are programmed into the μPDA in a learning/receive mode, and filed with user password protection. Once a correct password in entered, an icon-based menu is displayed on I/O area 316 in a user-friendly format. A master routine first queries a user for which device to access. For example, in a residential application, icons are displayed for such things as overhead garage doors, security systems, automatic gates, VCRs, television, and stereos.

In another aspect of the invention, a receiving station such as a host computer or peripheral interface has IR capabilities to communicate data directly from a nearby μPDA with an infrared interface. In a further embodiment the μPDA may interface in a cellular network and act as a wireless modem.

PERIPHERALS

A μPDA may serve as the platform for various peripheral attachments via expansion port 20 (Fig. 1B and others). Upon attachment to a peripheral, a dedicated pin or pins within expansion port 20 signal microcontroller 11, and a peripheral boot-strap application is executed. Interfacing control routines, which may reside in the peripheral or in the memory of the μPDA, are then executed, and the μPDA I/O interface displays the related menu-driven options after the linking is complete.

Scanner:

Fig. 14 is a plan view of a μPDA 10 with a scanner attachment 55 according to an embodiment of the present invention. The scanner attachment is assembled to the μPDA, making electrical connection via expansion port 20. In this embodiment the physical interface of the scanner is shaped to securely attach to the μPDA. Scanner attachment 55 has a roller wheel 57 or other translation sensor, which interfaces with wheel 18 of the μPDA, providing translation sensing in operation for the resulting hand-held scanner. In another aspect, scanner attachment 55 has a translation device which transmits the proper signal through expansion port 20. The scanner bar is on the underside, and one or more batteries 59 are provided within the scanner attachment to provide the extra power needed for light generation.

In the scanner aspect of the invention, scanner attachments 55 of different width D2 may be provided for different purposes. The bar may be no wider than the μPDA, or may be eight inches or more in width to scan the full width of U.S. letter size documents, or documents on international A4 paper. Unique control routines display operating information on the μPDA's I/O area 16 for scanning, providing a user interface for setup of various options, such as the width of the scanner bar, and providing identification for files created in the μPDA memory as a result of scan passes. Scanned data stored in the μPDA memory may be quickly transferred to the host via host interface 14 when the μPDA is docked. Unique routines may be provided to automate the process, so the user does not have to search for files and initiate all of the transfer processes.

Facsimile Option:

Fig. 15 is a plan view of a μPDA with a fax-modem module 89 attached according to an embodiment of the present invention. A fax and telecommunication capability is provided via conventional telephone lines to the μPDA by fax-modem 89 interfacing to expansion bus interface 20. The fax-modem has internal circuitry for translating from the bus states of the expansion bus to the fax protocol, and a phone plug interface 91. In another aspect, the μPDA can be docked in a host and be used in combination with fax-modem 89 to provide faxing and file transfers of both host and μPDA data files. In this case, the fax-modem routines are displayed on the host monitor.

Printer:

Fig. 16 is a plan view of a μPDA with a Centronics adapter interface according to an embodiment of the present invention. A

printer connector 93 engages expansion interface 20 by a connector 95 through a cable 97. Translation capability resides in circuitry in connector 93, which is configured physically as a Centronics connector to engage a standard port on a printer.

Barcode Reader and Data Acquisition Peripheral:

Fig. 17 is an isometric view of a μPDA 10 docked in a barcode reader and acquisition peripheral 100 according to an embodiment of the present invention. μPDA 10 is docked in docking bay 149. I/O interface 16 displays information through opening 147 according to specialized data acquisition applications. In this particular embodiment peripheral 100 has an IR interface 94, a microphone 103, a scanner port 101 (not shown), battery pack 105, and a numeric keypad pad 96 implemented as a touch-sensitive array.

Application routines enable the data acquisition peripheral to operate as, for example, a mobile inventory management device. The user may scan barcode labels with scanner 101 and enter information, such as counts, on keypad 96 or by voice input via microphone 103. Since applications of peripheral 100 are very specialized, only a limited voice recognition system is needed. The voice recognition system may prompt other command routines within the master applications as well.

As inventories are collected, the database may be displayed and also manipulated directly via I/O area 16 in open bay 147, or information may be downloaded at a prompt to a nearby host via IR interface 94.

Alternatively to frequent data transmission, data may be stored or an auxiliary option memory location in peripheral 100.

In another aspect, the data acquisition peripheral may be interfaced to the analog output of a monitoring device, such as a strip

chart recorder, and may digitize and store the incoming analog signals.

Solar Charger:

Fig. 18 is an isometric view of the side of a μPDA 10 opposite the I/O interface with a solar charger panel 98 according to an embodiment of the present invention. Panel 98 is positioned so that when μPDA 10 is in strong light, such as sunlight, the solar charger absorbs the solar energy and converts it to electricity to recharger battery 15 inside the μPDA. Solar charger 98 may be permanently wired to the circuitry of the μPDA or attached by other means and connected to a dedicated electrical port or the expansion port. The solar charger is placed so that the μPDA can be fully docked in a docking port with the panel in place. In another aspect, a detachable solar charger may be unplugged before docking the μPDA, and the detachable charger may then be of a larger surface area.

Games/Conference Center:

Fig. 19 is a largely diagrammatic representation of a Games Center unit 33 according to an aspect of the invention for connecting several μPDA units (37, 39, 41, and 43) together to allow competitive and interactive games by more than one μPDA user. Games Center unit 33 is controlled by an 80486 CPU in this particular embodiment. μPDAs may be connected to the central unit by cable connection via the expansion bus or the host interface of each μPDA, through a connector such as connector 35. The drawing shows four connectors, but there could be as few as two, and any convenient number greater than two.

As a further aspect of the present invention, the gaming center may serve as a conference center where a number of μPDAs may

exchange information. In this way, for example through custom routines stored and executable in central unit 33, a manager may update a number of salespeoples' μPDAs, including but not limited to merchandise databases, spreadsheets, price sheets, work assignments, customer profiles, address books, telephone books, travel itineraries, and other related business information while in conference.

Standard Keyboard:

Fig. 20 is an isometric view of a keyboard 151 connected by a cord and connector 153 to a μPDA 10 via the expansion port 20. In this example, the keyboard is a mechanical keyboard having a full-size standard key array and an on-board controller and interface for communicating with the μPDA. In other embodiments the keyboard may take many other forms, including a two-layer, flexible, roll-up keyboard as taught in U.S. Patent 5,220,521.

In addition to keyboards, other input devices, such as writing tablets and the like may also be interfaced to a μPDA via expansion port 20.

There are numerous additional ways to combine different embodiments of the μPDA for useful functions. For example, an IR-equipped μPDA attached to scanner 55 may transfer large graphic files in near real time to a host computer. If the files were of text, the host may further process the files automatically through an optical character recognition (OCR) application and send the greatly reduced ASCI files back to the μPDA. As discussed above, the μPDA family of devices establishes a protocol of software security and distribution as well as having the ability to be bus mastered by a host computer system for numerous applications.

A Host Interface Example

There are a number of different ways the host interface for the μPDA may be implemented. One way is according to a PCMCIA standard as described above. Another example for a host interface protocol for a μPDA host interface is described below, and referred to and described as an HCI bus, for High-Speed CPU Interconnect.

Fig. 21 is a block diagram of a computer architecture 211 according to the present invention. Paths 213 and 215 together represent an optimized High-Speed CPU Interconnect Bus called the HCI Bus by the inventors. The computer architecture based on the HCI bus is hereinafter the HCI architecture. It can be applied to PCs, Macs, workstations, as well as other types of computers, and is especially useful for portable computers such as notebook and palmtop computers.

Path 213 has 32 traces, and represents a multiplexed, address-and-data combination bus capable of conveying both 32 bit addresses and 32 bit data words. Path 215 has 10 traces for control signals further detailed below. The two paths together comprise the HCI Bus 240, a total of 42 traces, a considerably smaller number than most conventional bus structures, even multiplexed bus structures. The reduction in the number of traces results in a reduction in layout congestion, with a consequent reduction in the length of the traces. The shortened traces in turn reduce signal delays, so that the computer's performance is increased.

Peripheral I/O or expansion devices for the HCI computer architecture are represented by devices 217, 219, 221, and 223. They are shown coupled to HCI bus 240, preferably through expansion slots or connectors (not shown). VGA and Graphics processor card 217 includes high-speed video memory, and is responsible for displaying information on a connected monitor. SCSI and Ethernet interface card

219 provides coupling to various SCSI equipment, such as CD-ROM drives and tape backups, and to Ethernet local area networks. Multi-I/O card 221 is an interface for hard disks, floppy drives, fax/modems, keyboards, etc. Multi-media processor/DSP (digital signal processor) card 223 is a high-speed I/O and processor for audio and video signals, and can typically drive external speakers or audio amplifiers. Many other types of peripheral devices can be connected to HCI bus 240. Similar devices are widely available for most existing computer architectures. Suitable and relatively simple modifications can be made to make them compatible with the HCI Bus. The reduced number of traces in the HCI Bus in turn allow a reduction in the number of traces and connection contact pads on the peripheral devices.

Other than the peripheral interfaces, the basic computer in the HCI architecture is implemented in just four ICs. A system control chip 225 controls the bus, and contains the system clock (not shown). System memory 227 is high-speed, low-power, dynamic random access memory (DRAM). It communicates with the system controller through multiplexed addressing (MA) of standard rows address signals (RAS) and column address signals (CAS). 64-bit memory is used for better performance, although 32-bit memory can also be used. Data buffer 229 is the only buffer required in the system, because the bus can be driven by low power CMOS (complimentary metal-oxide semiconductor) outputs.

As an example specific to the buffer requirement, the conventional ISA bus is designed to support a TTL definition that is over a decade old. It requires a driving current of 24 ma multiplied by about 80 active signals, which amounts to about 1.6 amps, as described above. This large load requires, in most implementations, as many as twenty small buffer chips, or four or five large ones. The

overhead incurred in terms of bus length, heat, RFI, complexity, and so forth, is enormous. In contrast, the HCI architecture has virtually no DC loading and operates at about 10 milliamps per trace, multiplied by about 42 active traces switching at the same speed as most CPUs, or 16 to 33 MHz. As a result, only a single buffer is required in the HCI bus.

Computation in the HCI architecture, as in others, is handled by a CPU 231. The CPU is coupled to system controller 225 by a short 32-bit address bus 233, and to data buffer 229 by a short 32-bit data bus 235. Data is transferred between buffer 229 and DRAM 227 through short but separate address and data buses. Each of these buses has 36 signal paths, 32 of which are for address or data, and 4 are for control.

A significant feature of the HCI architecture is that it may be configured to support a large number of CPUs. Multiplexed buses have been designed in the past, such as the DEC TurboChannel, but these designs are compatible with only a single type of CPU or a single class of CPUs made by a particular manufacturer. In the HCI architecture, the CPU can be virtually any of the many available high-end microprocessors, such as the Intel 80486 and Pentium, DEC Alpha, Motorola 680x0, Sun SPARCs, and many others.

This important advantage of multi-CPU compatibility, or CPU independence, is provided by system controller 225. This chip has a programmable state machine for translating between the state dynamics of a variety of supported CPUs and the state dynamics of the HCI Bus. The programming can be done in a variety of ways, including EPROM, mask, RAM, etc. Even auto-programming can be achieved. In the case of auto-programming, the system controller will automatically recognize the CPU by reading the state, either pull-up or pull-down, of the CPU's output lines. The pattern of these outputs is

compared to a database of the patterns of known CPUs until a match is found. The system controller will then use the proper protocols to translate between the CPU and the HCI bus architecture.

In different embodiments of the invention, different types of programmability will be preferable. For example, on a MB designed to work with a variety of CPUs, an auto-programmable system controller will be preferred. In most cases, the HCI architecture will be designed to work with a single type of CPU, so that it is not necessary to change the setting. In these cases, mask programming is most cost effective.

Another feature of the HCI architecture is support for a high-speed master mode, programmed at the system controller chip. Each type of processor has its own algorithm for burst data transmission. In setting up the generic system controller to be compatible with a particular CPU, the system controller's master mode is set to be initiated and operated by the burst algorithm of the processor used.

Data transfer mechanisms of the architecture are CPU write to HCI, CPU read from HCI, HCI write to main memory, and HCI read from main memory.

Fig. 22 shows an alternative embodiment of the HCI architecture, comprising the architecture of Fig. 21 with additional, optional components. To increase performance, high-speed cache memory 237 can be coupled to the address and data buses from the CPU. It is controlled by the system controller. The cache need not be specifically designed to be compatible with HCI state dynamics, because it couples to the CPU bus instead of the HCI Bus.

Another option is a bus interface 239 for coupling the HCI bus to other bus structures, such as ISA, EISA, Microchannel, etc. The bus interface comprises an IC designed to convert between the state dynamics of the HCI bus and the dynamics of the bus to which the

HCI bus is to be coupled. For example, when a device on an ISA bus issues an interrupt, the devices on the HCI bus will not answer. In this case, bus interface 39 will become the default device which answers the interrupt.

Fig. 23 is a listing of the signals for the 42 lines of the HCI bus in the present embodiment. The pin numbers 0-41 in the left column are for reference only, and do not imply that the signals could not be in another order. The pins are used as follows:

Data0..31	are multiplexed with address2..31 & BE0..1
BE2..3	are remaining byte enables.
CLK	Bus clock.
ADS	Address strobe. This is used to start a cycle, and to indicate the burst length. ADS is low for one cycle for single word transfers. For burst transfers, it will remain low until the ready before the last word.
LDEV#	Device decode. This is used to indicate that a device is claiming a cycle. If no device pulls this line, the cycle will go to the default device, such as ISA bus interface 39. This signal is necessary for zero wait-state write support.
RDY#	Ready. Low indicates that valid read data is on the bus, or that a write word has been accepted.
R/W#	Read/Write. This is driven simultaneously with the address.
BREQ#	Bus request. A bus-mastering peripheral pulls low to request bus ownership. It is specific to each device.
BACK#	Bus acknowledge or grant. Goes low when bus

ownership is granted. It is specific to each device.

RESET# Low = system reset.

In conventional bus structures, in addition to the address, data, and control lines, many additional paths are needed for I/O and interrupt signals. An interrupt controller IC reads the IRQ signals, then triggers an interrupt in the CPU. This is a significant source of layout congestion in such architectures.

In the HCI bus, instead of issuing interrupts on dedicated paths, IRQs are mapped into high memory, as shown for example in the memory map in Fig. 24A. In this case, bus-mastering peripherals will first take control of the bus, then make master-mode memory writes to a predetermined block of addresses set aside for interrupts. Each interrupt, for example, from IRQ0 to IRQ15, is assigned a particular address. Also instead of using a separate IC for controlling interrupts, an interrupt controller 41 is built into system controller 225. The IRQ controller monitors the address space to trigger the appropriate interrupts. As a result, many paths and pins are saved.

Similarly, instead of communicating I/O on a dedicated path, it is also mapped into high memory. A small space, such as the 64K block from FFFE ' 0000 to FFFE ' FFFF, is set aside for I/O cycles. The mapping of I/O to high memory saves an additional one pin. The Intel CPU architecture has a separate I/O address space that can be accessed only by special instruction, such as IN, OUT, INS, and OUTS. This mapping will allow HCI-compatible devices to be used with other CPUs that do not have a separate I/O space.

Direct memory access (DMA) is not supported in the HCI bus. However, a "shadow" DMA controller 242 can be implemented by using just registers in the system controller to mimic an actual DMA

controller.

The calculated performance of the HCI bus is as follows: Maximum burst rate is 132 MBytes/sec at 33 MHz, if the memory system supports one cycle per word burst. HCI master write is accomplished in a minimum of 2 cycles. HCI master read is accomplished in a minimum of 3 cycles (or 4 cycles back-to-back). CPU write is a minimum 2 cycles (or 3 cycles back-to-back). CPU read is a minimum 3 cycles (or 4 cycles back-to-back).

Fig. 25 is a timing diagram showing the states of the Clock, HCI Bus Address/Data, Address Strobe, Device Decode, Ready, and Read/Write lines during back-to-back read operations. The first read operation is initiated when Address and Byte Enable bits (BE0, BE1, A2 .... A31 from lines 0 to 31 in Fig. 23) are enabled onto the bus after clock transition 241, simultaneously as the Address Strobe line is also enabled. These signal transitions are slightly delayed behind the clock transition that switches them (shown here as a quarter-cycle delay). The Read/Write line remains high throughout to indicate this is a read operation. The Address and Address Strobe signals remain asserted for one clock cycle and are disabled following clock transition 243. The one cycle Address Strobe indicates a single word transfer. After clock transition 243 the Device Decode line is asserted to indicate that a device is claiming the cycle. Data from the decoded address is enabled onto the bus following clock transition 245, while Ready is also enabled to indicate that valid data is on the bus. Ready and Data remain enabled for one clock cycle until they are disabled following clock transition 247. Dead cycles after clock transitions 243 and 247 prevent bus contention. An identical read cycle is initiated after clock transition 249.

Dead cycles, or wait-states, are required in the first cycle of a read, and in an address cycle following a read to prevent bus

contention. Change of bus ownership also requires a wait-state to avoid contention. Other operations, such as write cycles or following cycles in a burst, do not require wait-states.

Fig. 26 is a timing diagram showing the bus states for a burst-mode read operation, also called a HCI master-mode read. A burst-mode transfer is one in which a block of data is transferred in consecutive data cycles, but the address is transmitted only once at the beginning of the operation. Elimination of address cycles between each data cycle results in substantial throughput gains for bursts of even a few data words.

For clarity, Fig. 26 shows only two data cycles in a burst transfer. The different CPUs that can be used with the HCI Bus will differ in their burst transfer characteristics. In particular, they differ in the number of data words that can be transferred in a single burst, and in the counting sequence. For example, the Intel 80486 supports bursts of up to 4 words. System controller 225 will be suitably programmed to accommodate these differences.

As shown in Fig. 26, a burst-mode read operation begins following clock transition 241 with a bus state similar to the single-word read operation shown in Figure 5. The Address Strobe is enabled to start the Address cycle. Device Decode is enabled after the Address cycle to indicate address decoding is in progress. Unlike the single-word operation, the Address Strobe will remain enabled after the Address cycle until the ready before the last word, following clock transition 249 in this case, to maintain a burst. When data is being transferred following clock transitions 247 and 251, Ready is driven low to indicate that valid data is on the bus. Although a one clock cycle wait-state is shown between the data cycles, it is not necessary in this case. Successive data cycles in a burst can be done at zero wait-state.

Fig. 27 is a timing diagram showing the HCI bus states for back-to-back write cycles. As in Fig. 25, the Address Strobe is enabled for one cycle following clock transition 41, while an Address is also enabled onto the Address/Data lines. Unlike Fig. 25, the Read/Write line is enabled following clock transition 241 to signal a write operation. At the end of the address cycle, following clock transition 243, data is enabled onto the bus, while Device Decode is enabled to indicate that address decoding is in progress. Ready is driven low concurrently with the data cycle to indicate that valid data is on the bus. The write operation ends following clock transition 245, when the Address Strobe, Decode, Ready, and Read/Write lines are all returned to a high state.

Successive write operations repeat the states just described, one of which is shown following clock transition 247. A free cycle is shown between the two writes, from clock transition 245 to 247, but none is required.

Fig. 28 is a timing diagram showing the HCI bus states for a burst-mode write transfer. These states are similar to the states for a burst-mode read transfer, as shown in Fig. 26, for all lines except the Read/Write line, which is asserted following clock transition 241 to signal a write. Although wait-states are shown, they are not required. The first write data can come immediately after the address, as in Fig. 27.

The HCI bus described above is described with reference to various embodiments. One embodiment is as a host interface for a μPDA, as depicted in Fig. 6, item 26, in which case the interconnecting bus 26' for the host computer must be a compatible bus to the HCI bus described herein.

It will be apparent to one with the skill in the art that there are many changes that might be made and many other combinations that

might be made without departing from the spirit and scope of the invention. There are, for example, many ways to implement the support structure of the μPDA, and to interconnect the active components. One way has been illustrated by Fig. 22 and described in accompanying text. There are many alternatives to this preferred structure. There is also a broad range of sizes and form factors that might be assumed by devices according to the present invention. The use of well-known PCMCIA form factors has been disclosed, but other sizes and forms might also be provided in alternative embodiments. In larger embodiments, on-board peripherals may be implemented.

In addition to these alternatives, there are various ways the connectivity of a μPDA bus might be provided. The well-known PCMCIA standard has been disclosed as a preference, but other connectivity may also be used in alternative embodiments. Memory types and sizes may vary. Means of providing a security code may vary. The nature of the internal bus may vary. There are indeed many variations that do not depart from the spirit and scope of the invention.

What is claimed is:

1. A personal digital assistant module having an enclosure for enclosing and supporting internal elements, a microcontroller within the enclosure for performing digital operations to manage functions of the personal digital assistant module, a memory connected to the microcontroller by a memory bus structure for storing data and executable routines, a power supply within the enclosure for supplying power to functional elements of the personal digital assistant module, a display operable by the microcontroller and implemented on a surface of the enclosure, input apparatus connected to the microcontroller for providing commands and data to the personal digital assistant module, and a host interface comprising:

a host interface bus coupled to the microcontroller and to a first portion of a host interface connecter at a surface of the enclosure, the host interface providing address lines, data lines, and control signal lines, control signals on the control signal lines including read/write and at least one memory control signal, but no interrupt request (IRQ) signal or IRQ signal line; and

control circuitry for operating the host interface, including a stored list of addresses associated with specific interrupts;

wherein said control circuitry issues interrupts as addresses, each address associated with a specific interrupt in the stored list of specific interupts associated with addresses.

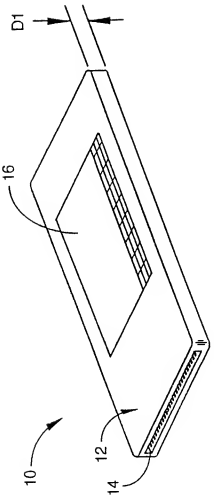


Fig. 1A

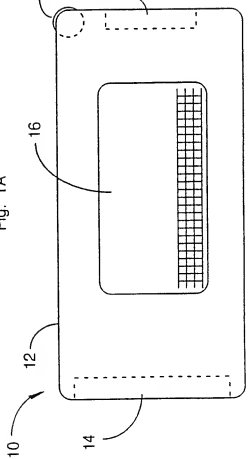


Fig. 1B

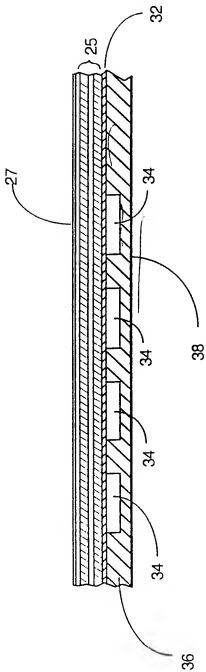


Fig. 2

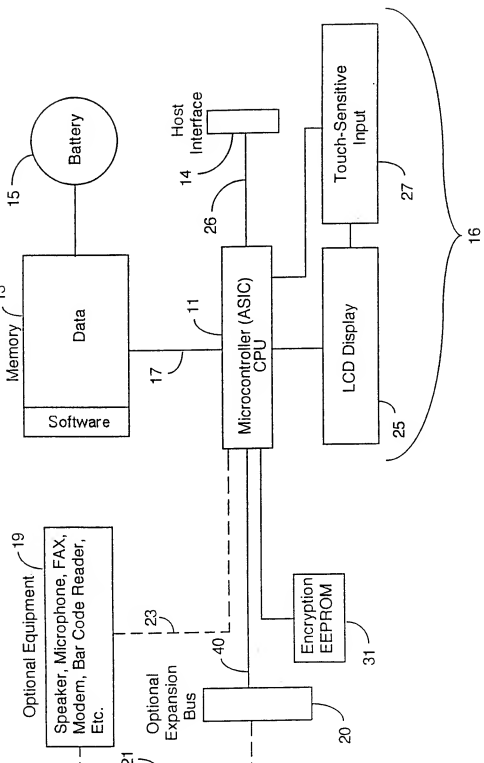

Battery
15
Host Interface
14
26
Touch-Sensitive Input
27
16
Memory
13
Data
Software
17
11
Microcontroller (ASIC) CPU
LCD Display
25
Optional Equipment
19
Speaker, Microphone, FAX, Modem, Bar Code Reader, Etc.
21
Optional Expansion Bus
23
40
20
Encryption EEPROM
31

Fig. 3

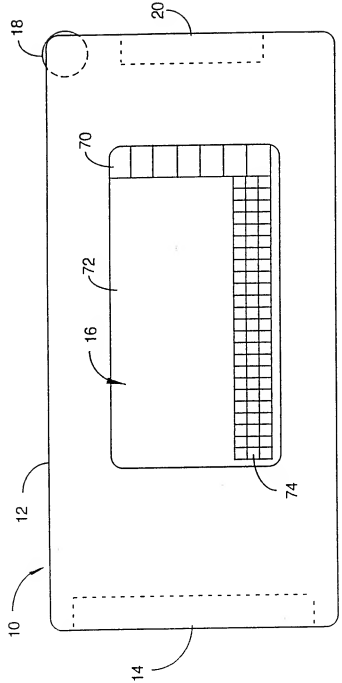


Fig. 4

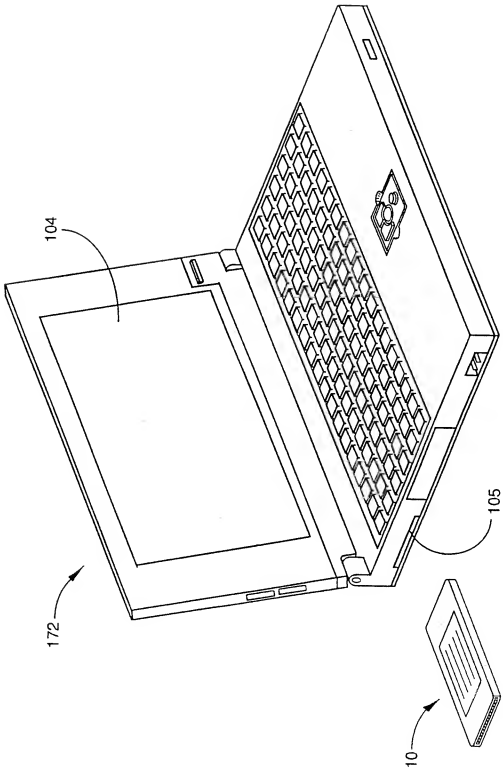


Fig. 5

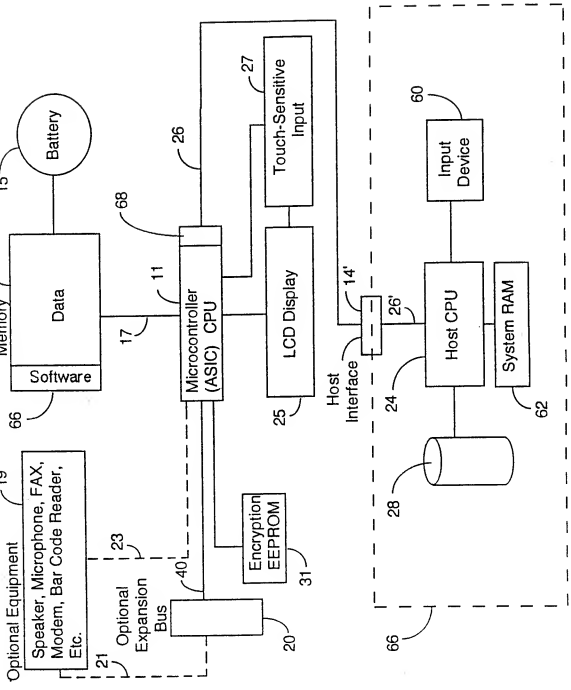

Optional Equipment
19
Speaker, Microphone, FAX, Modem, Bar Code Reader, Etc.
21
Optional Expansion Bus
23
40
20
Encryption EEPROM
31
Memory
13
Software
66
Data
17
Battery
15
68
Microcontroller (ASIC) CPU
11
26
LCD Display
25
Touch-Sensitive Input
27
Host Interface
14'
26'
24
28
66
Host CPU
System RAM
62
Input Device
60

Fig. 6

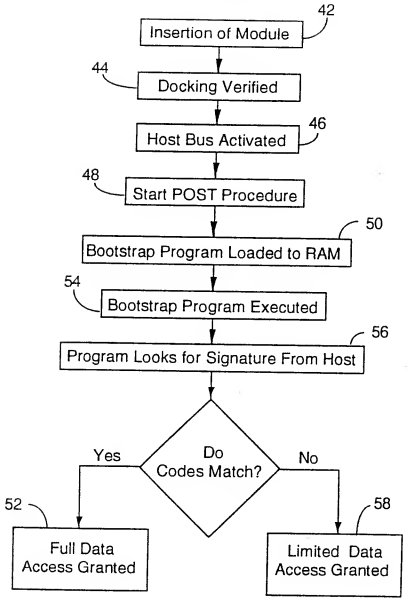

42
Insertion of Module
44
Docking Verified
46
Host Bus Activated
48
Start POST Procedure
50
Bootstrap Program Loaded to RAM
54
Bootstrap Program Executed
56
Program Looks for Signature From Host
Do Codes Match?
Yes
No
52
Full Data Access Granted
58
Limited Data Access Granted

Fig. 7

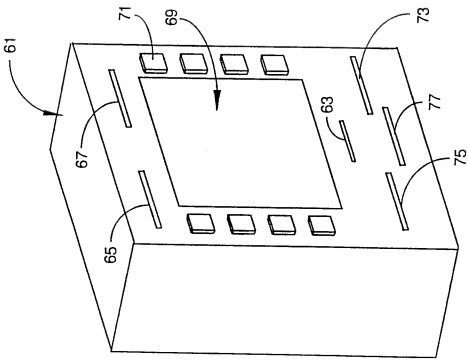


Fig. 8

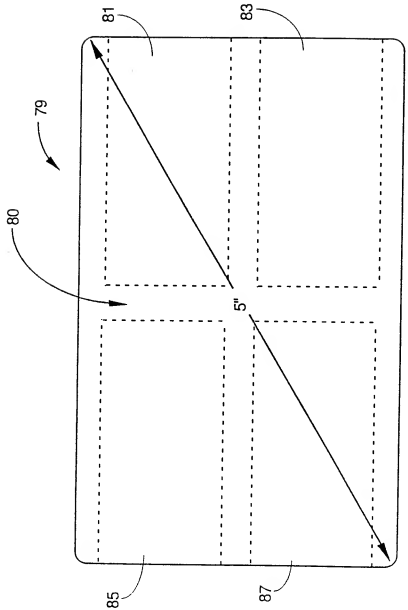


Fig. 9

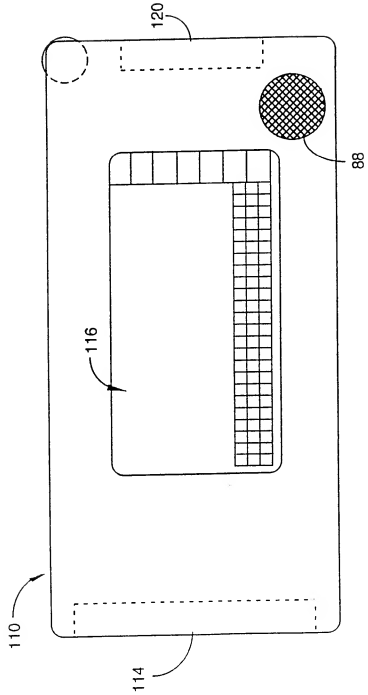


Fig. 10

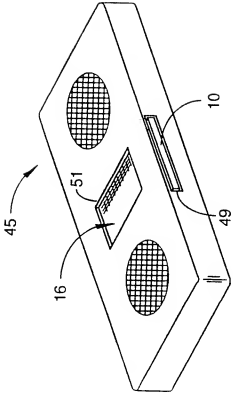


Fig. 11

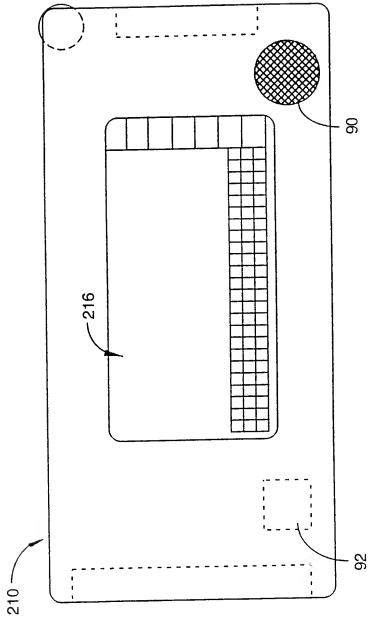


Fig. 12

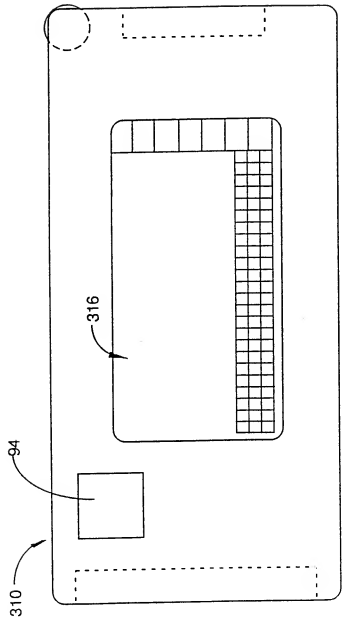


Fig. 13

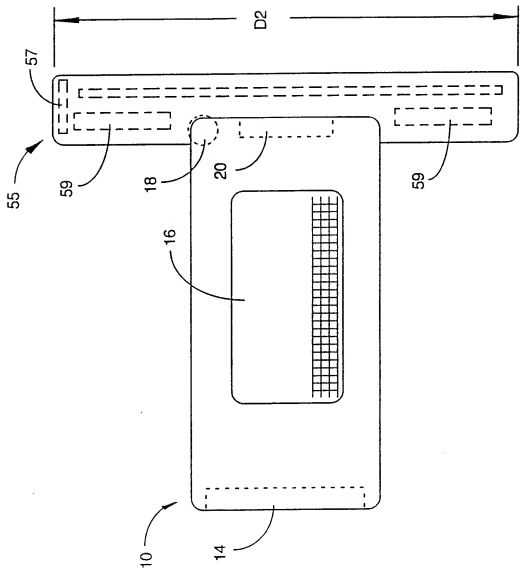


Fig. 14

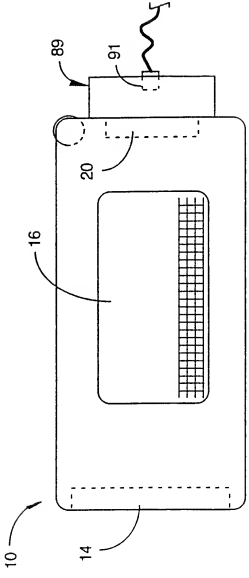


Fig. 15

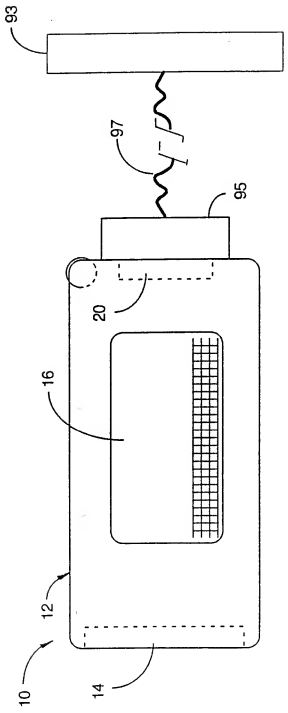


Fig. 16

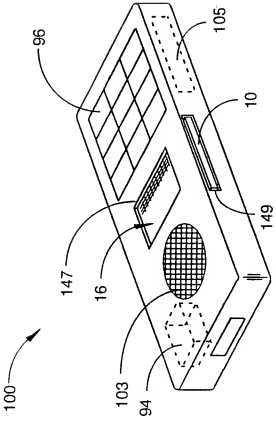


Fig. 17

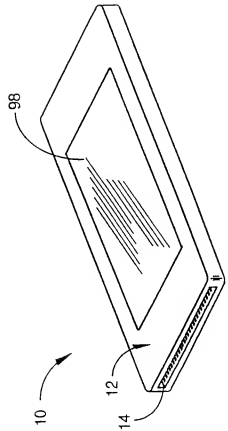


Fig. 18

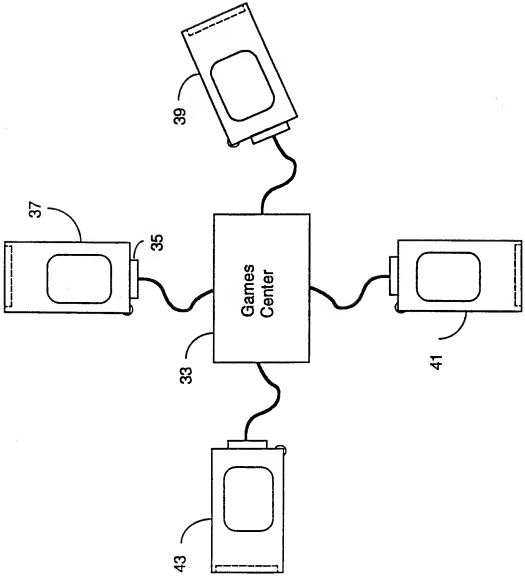


Fig. 19

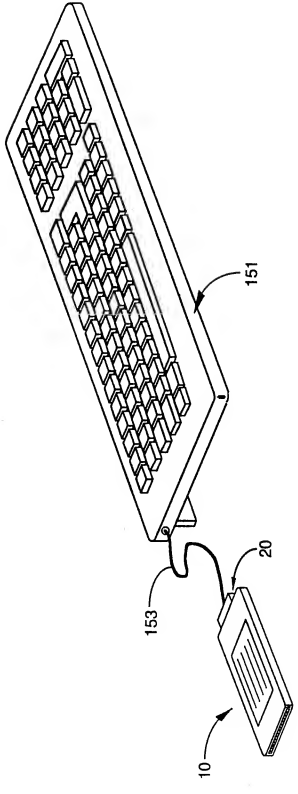


Fig. 20

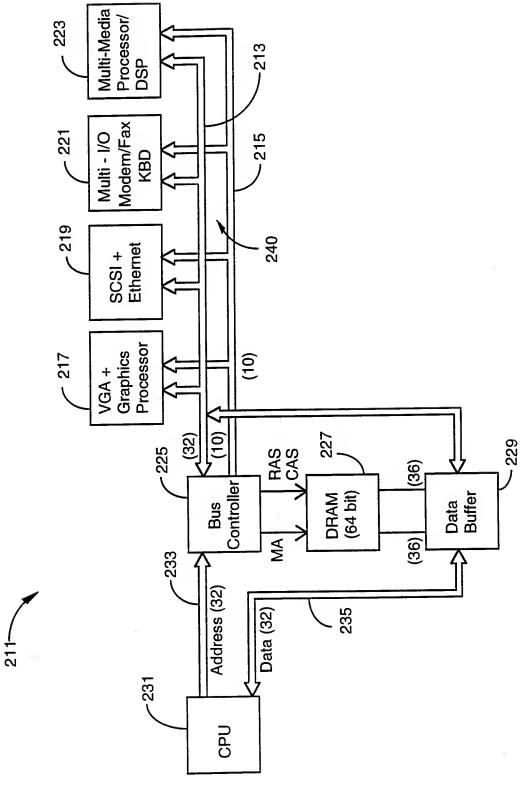

211
231
CPU
Address (32)
Data (32)
233
235
Bus Controller
225
(32)
(10)
RAS
CAS
MA
DRAM (64 bit)
227
(36)
(36)
Data Buffer
229
VGA + Graphics Processor
217
SCSI + Ethernet
219
Multi - I/O Modem/Fax KBD
221
Multi-Media Processor/ DSP
223
(10)
240
215
213

Fig. 21

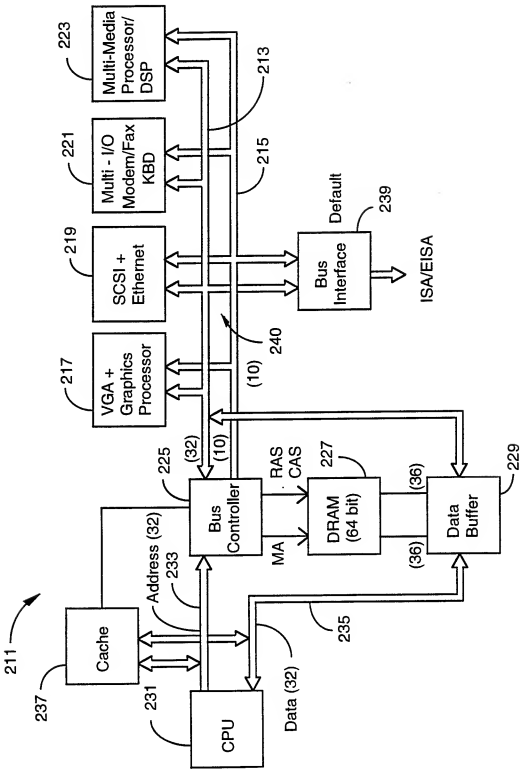

211
237
Cache
231
CPU
Address (32)
233
Data (32)
235
Bus Controller
225
(32)
(10)
MA
RAS
CAS
DRAM (64 bit)
227
(36)
(36)
Data Buffer
229
VGA + Graphics Processor
217
SCSI + Ethernet
219
Multi - I/O Modem/Fax KBD
221
Multi-Media Processor/ DSP
223
213
215
(10)
240
Bus Interface
Default
239
ISA/EISA

Fig. 22

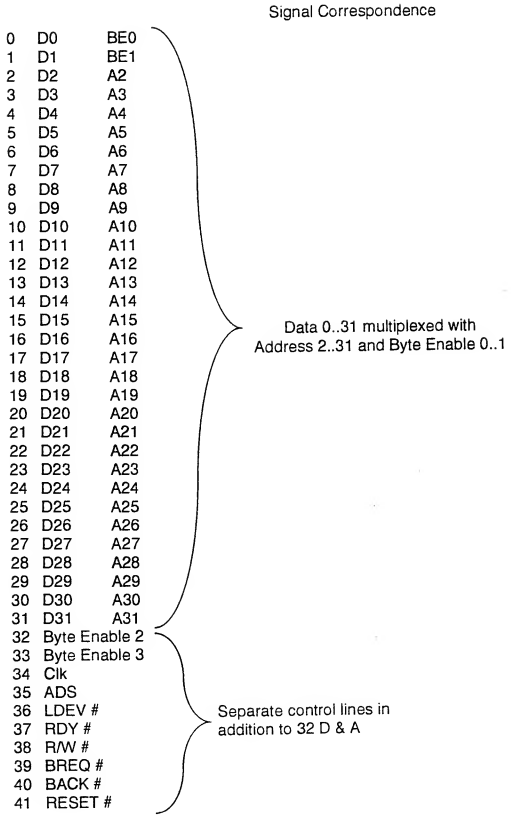

Signal Correspondence
0 D0 BE0
1 D1 BE1
2 D2 A2
3 D3 A3
4 D4 A4
5 D5 A5
6 D6 A6
7 D7 A7
8 D8 A8
9 D9 A9
10 D10 A10
11 D11 A11
12 D12 A12
13 D13 A13
14 D14 A14
15 D15 A15
16 D16 A16
17 D17 A17
18 D18 A18
19 D19 A19
20 D20 A20
21 D21 A21
22 D22 A22
23 D23 A23
24 D24 A24
25 D25 A25
26 D26 A26
27 D27 A27
28 D28 A28
29 D29 A29
30 D30 A30
31 D31 A31
Data 0..31 multiplexed with
Address 2..31 and Byte Enable 0..1
32 Byte Enable 2
33 Byte Enable 3
34 Clk
35 ADS
36 LDEV #
37 RDY #
38 R/W #
39 BREQ #
40 BACK #
41 RESET #
Separate control lines in
addition to 32 D & A

Fig. 23

FFFF'0000 - FFFF'FFFF	BIOS ROM (RESET VECTOR)
FFFE'0000 - FFFE'FFFF	I/O SPACE (64 KB)
FFFC'0000 - FFFD'FFFF	INTERRUPT ADDRESSES
	FFFC'0000 IRQ0
	FFFC'0004 IRQ1
	FFFC'0008 IRQ2
	....
	FFFC'003C IRQ15
0000'0000 - FFFB'FFFF	MEMORY SPACE

Fig. 24A

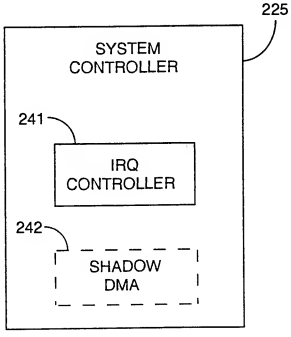


Fig. 24B

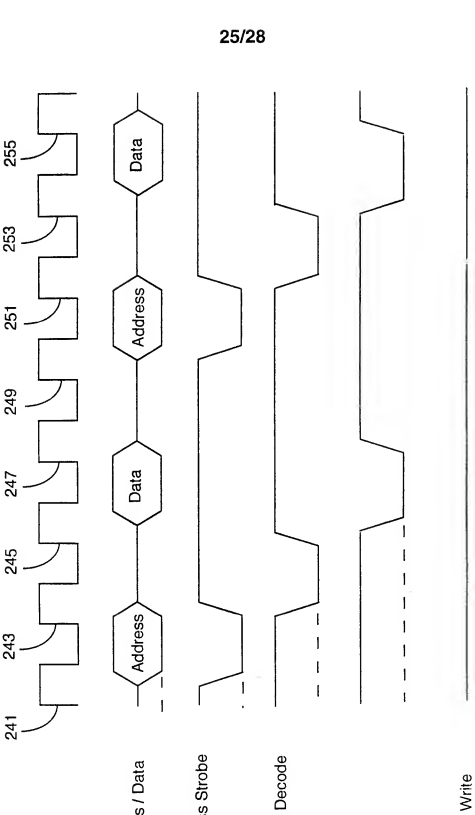

241
243
245
247
249
251
253
255
Clock
Address / Data
Address
Data
Address
Data
Address Strobe
Device Decode
Ready
Read / Write

Fig. 25

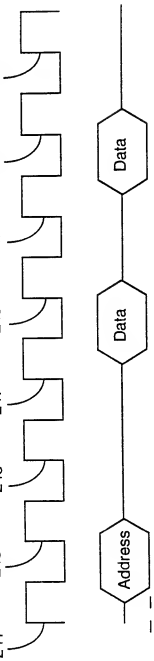

241
243
245
247
249
251
253
255
Clock
Address / Data
Address
Data
Data
Address Strobe
Device Decode
Ready
Read / Write

Fig. 26

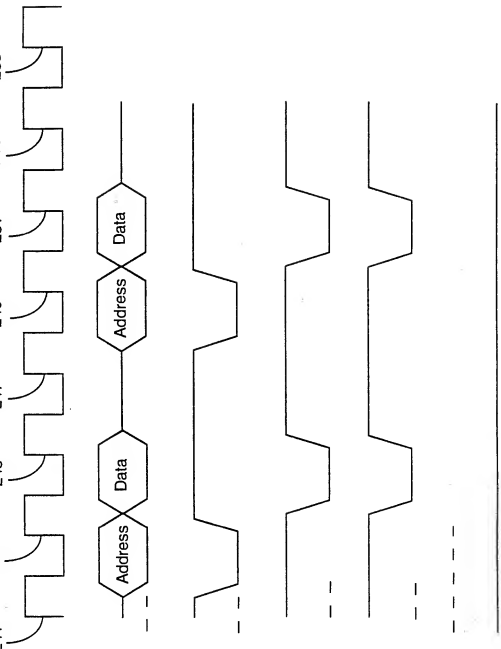

241
243
245
247
249
251
253
255
Clock
Address / Data
Address
Data
Address
Data
Address Strobe
Device Decode
Ready
Read / Write

Fig. 27

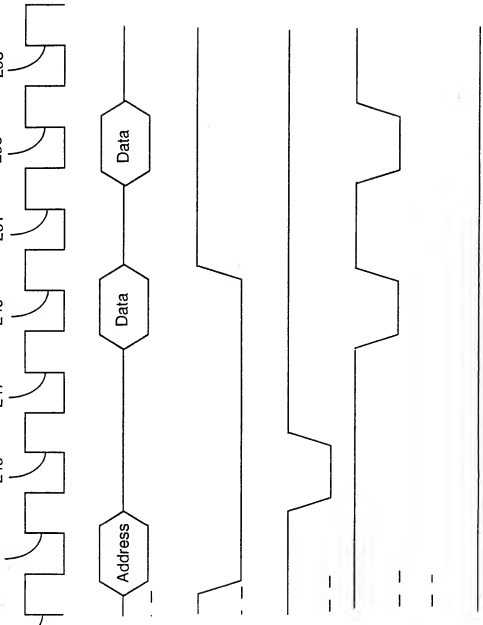

241
243
245
247
249
251
253
255
Clock
Address / Data
Address
Data
Data
Address Strobe
Device Decode
Ready
Read / Write

Fig. 28

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US95/08603

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) :G06F 13/00, 13/24, 15/16
US CL :Please See Extra Sheet.
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)
U.S. : 395/325, 800, 275, 425, 725, 155; 345/160, 169, 173; 379/93, 96; 364/709.09; 320/21

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
APS: bus?, interrupt#, (bus controller#), (interrupt controller#), (digital assistant#), interfac?, (enclosure# or cover#), (microcontroller# or micro(w)controller#)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US, A, 5,212,800 (MENSCH, JR.) 18 May 1993, Fig. 1 and col. 2, lines 6-62.	1
Y	US, A, 5,280,621 (BARNES ET AL) 18 January 1994, Fig. 2 and col. 2, lines 3-17.	1
Y	US, A, 5,247,685 (LANDRY ET AL) 21 September 1993, Fig. 3 and col. 1, line 31- col. 2, line 2.	1
Y	US, A, 5,040,111 (AL-SALAMETH ET AL) 13 August 1991, Fig. 1 and col. 1, line 40-col. 2, line 7.	1
Y	US, A, 5,265,255 (BONEVENTO ET AL) 23 November 1993, Fig. 3 and col. 2, lines 18-32.	1
Y	US, A, 4,159,516 (HENRION ET AL) 26 June 1979, Fig. 1 and col. 2, line 23-col. 3, line 2.	1

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be part of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
15 AUGUST 1995	04 OCT 1995
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer Gopal C. Ray Telephone No. (703) 305-9647

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US95/08603

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US, A, 5,237,692 (RAASCH ET AL) 17 August 1993, Fig. 1 and col. 2, line 50-col. 3, line 48.	1
Y	US, A, 5,218,686 (THAYER) 08 June 1993, Fig. 1 and col. 1, line 61-col. 2, line 12.	1
Y	US, A, 5,109,493 (BANERJEE) 28 April 1992, Fig. 2 and col. 2, line 49-col. 3, line 29.	1

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US95/08603

A. CLASSIFICATION OF SUBJECT MATTER:
US CL :

395/325, 800, 275, 425, 725, 155; 345/160, 169, 173; 379/93, 96; 364/709.09; 320/21